

Мироненко О.В., Романов Д.А., Серебренников В.А.
УНИВЕРСАЛЬНОЕ АВТОМАТИЗИРОВАННОЕ РАБОЧЕЕ
МЕСТО ДЛЯ СОЗДАНИЯ АППАРАТНО-ПРОГРАММНЫХ
МОДЕЛЕЙ СИСТЕМ ЦИФРОВОЙ ОБРАБОТКИ
СИГНАЛОВ

s_no_w@e1.ru

*ФГАОУ ВПО «Уральский федеральный университет имени первого
Президента России Б.Н. Ельцина»
г. Екатеринбург*

В статье описывается автоматизированное рабочее место (АРМ), разработанное для организации практических работ по курсу «Цифровые сигнальные процессы». В статье описывается структура АРМ, организация протокола связи между программной средой разработки цифровых устройств и микропроцессорным модулем, а также интерфейс пользователя. Приводятся перспективы использования АРМ. Статья полезна для студентов старших курсов, магистрантов и преподавателей, занятых в области цифровой обработки сигналов.

Mironenko O.V., Romanov D.A., Serebrennikov V.A.
UNIVERSAL WORKBENCH FOR CREATION HARDWARE AND
SOFTWARE MODELS OF DIGITAL SIGNAL PROCESSING SYSTEMS

The paper describes the workbench created for the practical training courses named "Digital signal processors". This paper is reported about workbench structure, interface protocol between digital devices development software and microprocessor unit and also said about user interface. In conclusion, prospects of using workbench are presented. The paper is useful for undergraduates and professors, which deals with digital signal processing.

Актуальность проекта

В рамках специальности «Проектирование и технология электронных вычислительных средств» проводится подготовка специалистов, владеющих методами проектирования, изготовления и эксплуатации современных радиоэлектронных и компьютерных систем. Современные радиоэлектронные системы являются сложными системами. Наряду с аппаратной реализацией, они содержат и программные алгоритмы, в том числе алгоритмы цифровой обработки сигналов и цифрового управления, реализуемые при помощи различных микропроцессорных систем. Поэтому в программу подготовки специалистов были включены курсы «Микропроцессорные системы управления» и «Цифровые сигнальные процессоры». Лекционная часть курсов знакомит студентов с основными методами цифрового управления и обработки сигналов, которые в дальнейшем должны быть реализованы практически.

В число самостоятельных работ входят лабораторные работы, курсовое проектирование, учебно-исследовательские работы, направленные на самостоятельное создание студентами аппаратно-программных моделей реальных систем цифровой обработки информации и сигналов (ЦОИС). Основным преимуществом таких работ является наглядность полученных результатов, а также приобретение навыков работы по проектированию реальных систем, начиная с этапа эскизного проектирования.

Для того чтобы организовать, нужны универсальные автоматизированные рабочие места (АРМ). Универсальность заключается в возможности использования рабочего места для реализации различных задач, а именно:

- создание и оптимизация структурно-функциональной схемы устройства с выделением аппаратной (HW) и программной части (SW);
- моделирование устройств ЦОИС с использованием стандартных пакетов (MatLab и LabView) и выбор модуля с микроконтроллером или цифровым сигнальным процессором для аппаратной реализации;
- программирование и автономная отладка программ (ЦОИС) для выбранного типа МК, ЦСП с использованием реального модуля с этим процессором и систем разработки и отладки программ на PC (IDE);
- создание аппаратно-программной модели с имитацией внешних систем на PC (SW, HW). При этом для связи с базовым модулем используются модули, подключаемые к PCI (модули фирмы National Instruments NI), и последовательные порты PC (USB, COM,...), а также пакет LabView;
- комплексная отладка SW и HW с использованием созданной модели.

Описание автоматизированного рабочего места

Автоматическое рабочее место (далее АРМ) включает аппаратные и программные средства для разработки ядра системы и аппаратно-программного моделирования внешнего окружения.

Структурная схема аппаратной реализации АРМ показана на рис. 1.

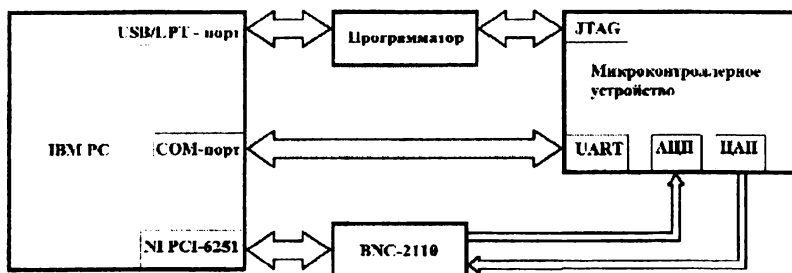


Рис. 1. Структурная схема аппаратной реализации АРМ

В данной схеме микроконтроллерное устройство – это отладочный модуль одной из ведущих фирм разработчиков, содержащий микроконтроллер (МК) или цифровой сигнальный процессор (DSP) и предназначенный для освоения технологий аппаратно-программного проектирования систем на основе этого МК, DSP. Такие модули и программное обеспечение к ним поставляются фирмами по университетским программам на льготных условиях.

На данный момент создано АРМ на основе модулей с микроконтроллером LPC2368 фирмы NXP и MSP430F149 фирмы Texas Instruments). По средствам USB или LPT порта через программатор, соединенный с микроконтроллерным устройством по интерфейсу JTAG, в микроконтроллерное устройство из IBM PC (персональный компьютер) записывается исполняемая программа и производится ее отладка в случае необходимости. COM-порт на стороне персонального компьютера и UART на стороне микроконтроллерного устройства соединены между собой по интерфейсу RS-232. По средствам этого интерфейса контроллер и компьютер обмениваются между собой данными, которые необходимы для управления микроконтроллерным устройством. Также в персональном компьютере имеется устройство сбора данных, в нашем случае это DAQ устройство, представляющее собой модуль фирмы National Instruments NI PCI-6251, который подключается в слот PCI. К ней через 68-контактный разъем подключается вспомогательный соединительный модуль BNC-2120. В свою очередь, этот модуль соединен двумя экранированными кабелями с АЦП и ЦАП микроконтроллерного устройства. В общем случае к АЦП микроконтроллерного устройства может быть подключен любой генератор сигнала с согласованной амплитудой входного сигнала микроконтроллерного устройства, а к ЦАП – можно подсоединить регистрирующее сигналы устройства (например, осциллограф).

Структурная схема программной реализации АРМ показана на рис. 2. На стороне IBM PC с использованием IDE (Integrated Development Environment – интегрированной среды разработки) пишется программа на микроконтроллер (Program Data) и производится отладка этой программы. Для каждого микроконтроллера используется чаще всего своя конкретная IDE, например, для LPC2368 была использована IDE KEIL uVision v.4, а для микроконтроллера MSP430F149 использовалась IAR Embedded Workbench. В программе зашитой в микроконтроллере и в виртуальном приборе, созданном в программе LabView, организован протокол обмена данными (эти данные условно назовем I/O Control Data) между контроллером и виртуальным прибором. Также в виртуальный прибор, созданный в LabView, может генерировать создаваемый нами сигнал и выдавать его в контроллер на АЦП (принятый контроллером аналоговый сигнал и представленный в цифровой форме назван на схеме Input ADC Data) и считывать аналоговый сигнал с ЦАП микроконтроллера (данные, которые идут на ЦАП контроллера, названы Output DAC Data).

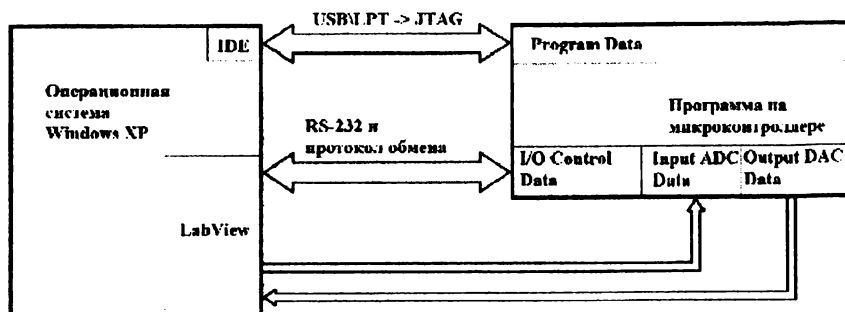


Рис. 2. Структурная схема программной реализации АРМ.

Основная задача виртуального прибора на LabView – это реализация протокола обмена данными между контроллером и ПК на стороне ПК. Дополнительно виртуальный прибор может генерировать сигнал и принимать аналоговый сигнал в оцифрованной форме, а также синтезировать коэффициенты БИХ и КИХ фильтров для дальнейшей передачи этих коэффициентов в контроллер.

Основная задача программной части на контроллере – это поддержание протокола обмена данными между ПК и контроллером со стороны контроллера, принятие и анализ управляющих данных (I/O Control Data) и реализация моделирования методов цифровой обработки сигналов. Далее приведено пояснение к типам данных, приведенных на структурной схеме программной реализации.

Program Data – данные самой программы, которые прошиваются в контроллер.

Input ADC Data являются отсчетами оцифрованного входящего аналогового сигнала и подвергаются дальнейшей обработке (фильтрации, модуляции, демодуляции и т. д. в зависимости от внутреннего состояния программы контроллера).

Output DAC Data являются отсчетами выходного обработанного сигнала, подаются на ЦАП.

I/O Control Data – управляющие данные, полученные от ПК, которые несут команду для контроллера, и данные, сопутствующие команде (например, команда, которая приводит контроллер в состояние работы как БИХ фильтра, а данные – это коэффициенты для этого фильтра), также эти данные могут быть посланы на ПК из контроллера, чтобы сообщить какие-либо сведения, предусмотренные программой.

Программа для контроллера на данном этапе имеет два внутренних состояния, соответствующие работе контроллера как БИХ и КИХ фильтра. Программа устроена таким образом, что в нее легко можно добавлять новые внутренние состояния, нужно лишь в одну функцию добавить код, который будет

интерпретировать полученные данные от ПК, а в другую функцию добавить код, который будет выполнять программа в новом внутреннем состоянии.

Для реализации связи между компьютером и внешним устройством требовалось выбрать как физический интерфейс (RS-232, RS-485, USB и т.д.), так и логический протокол обмена данными. Существует большое количество протоколов обмена данными, но они могут быть избыточны для реализации простыми микроконтроллерными устройствами и расходовать вычислительные мощности микроконтроллера устройства в ущерб «полезным» вычислениям.

В данной работе был создан простой протокол для реализации связи между персональным компьютером и внешним микроконтроллерным устройством. Основой протокола является протокол SLIP (Serial Link Interface Protocol). Передача данных осуществляется в двоичном виде, т.е. используются все возможные значения байта (00h...FFh). Для передачи служебной информации зарезервированы три управляющих кода: FBEG = 0xC0 (Frame Begin), FEND = 0x0C (Frame End) и FESC = 0xBD (Frame Escape) (табл. 1). Управляющий код FBEG служит для обозначения начала посылки, FEND - конец посылки, а код FESC служит для передачи ESC-последовательностей. Если в потоке данных встречаются байты, значения которых совпадают с управляющими кодами, производится подмена этих байт ESC-последовательностями. Такой механизм называют байт-стаффингом (byte stuffing). Управляющий код заменяется последовательностью <FESC>, <(Управляющий код)XOR(FXORESC)>. Где <(Управляющий код)XOR(FXORESC)> означает, что данный двоичный символ получен при операции XOR («сложение по модулю два») управляющего кода и символа FXORESC = 0x55 (табл. 2).

Структура пакета в протоколе следующая: пакет всегда начинается управляющим кодом FBEG и заканчивается кодом FEND. Все, что находится между этими кодами, назовем телом пакета. Первый байт тела пакета содержит количество информационных байт, которое требуется отправить. Минимальное количество информационных байт, которые содержит пакет, равно 0, максимальное 255. Перед отправкой тело пакета кодируется, если это требуется ESC-последовательностями, и обрамляется по краям кодом начала и конца пакета соответственно.

Протокол является логическим уровнем интерфейса управления оборудованием с помощью асинхронного последовательного канала. Физический уровень интерфейса протоколом не определяется, может использоваться, например, RS-232, RS-485 или USB. Протокол позволяет производить обмен данных длиной до 255 байт. В нашем случае в качестве физического уровня интерфейса выступает интерфейс RS-232, настроенный следующим образом: число бит в посылке – 8, количество стоп-бит – 1, бит четности – нет, скорость обмена – 9600 бод.

Таблица 1

Управляющие коды протокола

Обозначение	Пояснение	Значение
FBEG	Frame Begin (код начала пакета)	0xC0
FESC	Frame End (код окончания пакета)	0x0C
FEND	Frame Escape (код ESC-последовательности)	0xBD
FXORESC	Frame XOR Escape (код для кодирования байт)	0x55

Таблица 2

Подмена байт данных ESC-последовательностями

Байт дан-ных	(Байт данных)XOR(FXORESC)	Передаваемая последо-вательность
0xC0	$(0xC0) \oplus (0x55) = 0x95$	0xBD, 0x95
0x0C	$(0x0C) \oplus (0x55) = 0x59$	0xBD, 0x59
0xBD	$(0xBD) \oplus (0x55) = 0xE8$	0xBD, 0xE8

Для разработки фильтра необходимо в пользовательской программе задать характеристики фильтра (тип фильтра, частоты среза, порядок и др.), затем оценить полученный фильтр, сравнив его амплитудно-частотную характеристику с желаемой. Если фильтр является искомым, то нажимаем клавишу «Send information» и коэффициенты загружаются в микроконтроллер. После этого можно подавать сигнал на входной аналого-цифровой преобразователь (АЦП) микроконтроллера и снимать сигнал с цифро-аналогового преобразователя (ЦАП).

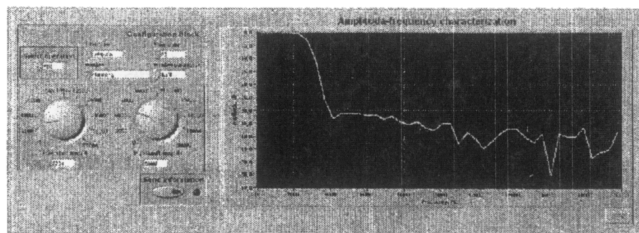


Рис. 3. Окно пользовательской программы

Перспективы использования проекта

АРМ разработано с целью обучения студентов практическим навыкам разработки алгоритмов цифровой обработки сигналов, в частности БИХ и КИХ фильтров. Однако ввиду универсальности рабочего места оно может быть использовано для различных целей. Ниже приведены возможные практические применения устройства.

1. Поскольку алгоритм реализации цифровой обработки сигналов является доступным и описанным на стандартном языке (язык программирования

Си), то возможно расширение существующего набора методов цифровой фильтрации в данном устройстве (например, разработка алгоритмов модуляции и демодуляции сигнала, медианной фильтрации, адаптивной фильтрации и др.). Это реализуется путем добавления соответствующего кода в программу и может быть использовано в качестве обучения студентов навыкам написания алгоритмов обработки сигналов на реальном устройстве и в дальнейшем проработке этих методов на лабораторных занятиях.

2. Рабочее место может быть использовано для обработки сигнала, сохраненного в файле, либо взятого с реального устройства. Поэтому оно может применяться для сравнения различных методов обработки для реальных сигналов и разработки реальных систем цифровой обработки сигнала. Обработанный сигнал может наблюдаться как на ПК с использованием соответствующих модулей, так и на осциллографе. Частота реального сигнала должна быть приведена к соответствующей частоте дискретизации устройства.
3. Использование стандартного интерфейса и протокола обмена данными позволяет использовать в установке различные микропроцессорные модули (например, модуль фирмы Texas Instruments на основе DSP TMS320C2812 либо модуль фирмы NXP на основе микроконтроллера на базе ядра Cortex_M3). Это позволяет сравнивать производительность микроконтроллеров различных семейств, а также производить цифровую обработку сигналов в реальном времени, что увеличивает возможности данного АРМ.

Некрасов А.С., Зайдуллина С.Г.

**РАЗРАБОТКА ЭЛЕКТРОННЫХ ОБУЧАЮЩИХ
ИГРОВЫХ ПРИЛОЖЕНИЙ**

konk_image@mail.ru

*ГОУ ВПО «Башкирский государственный педагогический университет
им. М. Акмуллы»
г. Уфа*

Статья посвящена разработке обучающих игровых приложений.

Zaydullina S.G., Nekrasov A.S.

DEVELOPMENT OF ELECTRONIC LEARNING GAME APPLICATIONS

The article is devoted to the development of educational gaming

В современной педагогике происходит постепенная замена традиционной парадигмы образования, основанной на пассивном усвоении знаний на